FROM GPS TRACKS TO CONTEXT

Inference of high-level context information through spatial clustering

Adriano MOREIRA, Maribel Yasmina SANTOS

Location-aware applications use the location of users to adapt their behaviour and to select the relevant information for users in a particular situation. This location information is obtained through a set of location sensors, or from network-based location services, and is often used directly, without any further processing, as a parameter in a selection process. In this paper we propose a method to infer high-level context information from a series of position records obtained from a GPS receiver. This method, based on a spatial clustering algorithm, automatically estimates the location of the places where a user lives and works. The achieved results show that the proposed approach rapidly converges to the real locations, and that the proposed algorithm can be used to simultaneously estimate both the home and workplace, while adapting to a wide range of spatio-temporal human behaviours.

KEYWORDS

Location context, Location-aware applications, Spatial clustering, Inference, GPS tracks.

INTRODUCTION

Since Weiser [15] published his visionary work in 1991, many context-aware and location-based systems have been developed. As a new research area, many of the first results were achieved through simple implementations where location and other contextual data were used directly from sensors to provide adaptation or selection of information accordingly to the user context. In most of the initial context-aware systems, location was used as the main dimension of context, thus leading to the concept of location-aware or location-based systems. While the first location-aware prototype systems worked only in very controlled environments and, quite often, providing a single function, the recent advances in sensors technology and wireless networks provide a new and rich environment for the deployment of sophisticated context-aware systems. In particular, the wide spread of mobile cellular telephony networks and wireless local area networks is enabling the deployment of more advanced context-aware applications. Current location services in mobile cellular networks provide the position of users, with increasing resolution, in a geographic referential system such as the WGS84 datum. In addition, an increasing number of vehicles such as cars, taxis and trucks, are equipped with Global Positioning System (GPS) receivers as part of their navigation or tracking systems. All these infrastructures facilitate the access to positioning information.

On the other hand, positioning sensors and positioning services only provide positioning information and do not describe the situation of the user. Although the position by itself can be used directly by many applications, such as navigation systems, many other dimensions of context can be used by applications to better adapt their behaviour to the user needs [2, 14]. Moreover, in addition to context information acquired from a variety of sensors, some other context information can be calculated and estimated from this raw data [8, 10-13].

These calculated or estimated context dimensions can also be used as cues to infer high-level information about the user context [8, 11, 12] which can be very useful for a wide range of context-aware applications. In particular, more sophisticated context-aware applications can benefit from richer descriptions of location. These references to a location, that describe the context of the users, use high-level descriptions of location (e.g. street name, town, home, workplace, etc.) instead of numerical descriptions such as a pair of coordinates or the CellID in a cellular network.

Therefore, one step towards the wide adoption of context-aware systems is the use of context models that are close to the way people behave and interact. High-level location information can contribute to this goal.

With the work reported in this paper, we aim to contribute to the enhancement of the user context by estimating new high-level parameters related to location that can be used by applications to better adapt their behaviour to the user situation, and which descriptions human beings understand. In particular, we aim to automatically estimate the places where a user lives and works from a set of position readings obtained from position sensors. This new information (position of the home and workplaces of a user) can be used by many context-aware applications as a cue to infer if the user is working or not (activity), if the user is at a familiar place or not, to trigger notifications remembering the user of specific tasks when entering or leaving these places, or to select applications that are relevant for a particular place.

Our approach, based on a spatio-temporal clustering algorithm, estimates the position of the places where a user lives and works, classify these places automatically as "home" or "workplace", and provides a measure of the confidence on the estimated positions. To estimate these parameters, we sequentially process a series of data records representing the sequence of positions visited by a mobile user and, through a new clustering algorithm, we estimate where the user lives and works. This estimation process is completely automatic and is not assisted by the user.

In the next sections we describe our approach to estimate the "home place" and "workplace". This description is preceded by a short introduction to clustering algorithms. The achieved results are then presented and discussed. This is followed by an analysis of the generality and validity of the proposed algorithm. Finally, we present our conclusions and describe some of the open issues that require further work.

CLUSTERING ALGORITHMS

Clustering is a discovering process that groups a set of data objects in a way that maximises the similarity between the objects inside a cluster, and minimise the similarity between different clusters [6, 7]. It is considered a data mining technique and an unsupervised learning technique since the user has no influence in the discovery process [5].

Spatio-temporal datasets present a characteristic that make them different from spatial datasets - the changes verified in the spatial data can be continuous, like the position of a moving object. Spatio-temporal data mining refers to the extraction of implicit knowledge, spatial and temporal relationships, or other patterns not explicitly stored in spatio-temporal databases [16].

Spatial data clustering allows the identification of clusters, or densely populated regions, according to a specific distance measurement in datasets [6]. Some popular clustering algorithms like *k-means* or *k-medoid* have several drawbacks when applied to large databases, assuming that all objects to be clustered can reside in main memory at the same time, and they are too inefficient on large databases [3]. In spatio-temporal clustering, the time dimension constitutes one of the characteristics to be considered when looking for similarities between objects.

Some of the well-known types of clustering algorithms are partitioning, hierarchical and density-based [6]. Partitioning algorithms identify a partition of a database, assigning its *n* objects to a set of *k* clusters where *k* is an input parameter. Each cluster is represented by the gravity centre (centroid) of the cluster (*k-means*) or by one of the objects of the cluster located near to its centre (*k-medoid*) [3]. In these algorithms, each object is assigned to the closest representative cluster (the cluster with the minimum measurement value).

Hierarchical clustering algorithms create a hierarchical composition of a given set of data objects. They can be agglomerative or divisive, based on how the hierarchical composition is performed. In

hierarchical methods, once a step (merge or split) is done, it can never be undone, which does not allow the correction of erroneous decisions [6].

Density-based algorithms are based on the notion of density and not only on the distance between objects. Their assumption is to continue growing the given cluster as long as the density, represented by the number of data points, exceeds some threshold. For each data point within a cluster, the neighbourhood of a given radius has to contain at least a minimum number of points. These algorithms can be used to filter out noise and discover clusters of arbitrary shapes [6]. Typically, these algorithms identify clusters that are dense regions of objects, which are separated by regions of low density representing noise. DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [4] is one example of a density-based algorithm. It is based on the key idea that for each point in a cluster, the neighbourhood of a given radius has to contain a minimum number of points (the density in the neighbourhood has to exceed some threshold). It requires two input values, the minimum distance between two points and the minimum number of points that the neighbourhood of a given point must have.

Independently of the adopted approach, partitioning, hierarchical or density-based, the mentioned algorithms require the availability of all objects in the dataset to perform the clustering. However, there are some application domains in which it is not possible to use the entire dataset in the clustering process - the information is not all available at some specific point in time, or for privacy reasons it should not be recorded. This is the case when data is collected in real-time and results of the clustering process are needed as the data is becoming available. For these reasons, this paper proposes a new clustering algorithm that processes the data as it is collected in real-time. This new algorithm integrates assumptions from the several clustering approaches described above. It starts with a set of *N* given clusters, used to estimate the home and workplace positions of the user. At the end of the estimation process, each of these clusters has associated a confidence metric that represents the probability of being the right place. This algorithm is described in detail in the following section.

"HOME PLACE" AND "WORKPLACE" ESTIMATION

For estimation of the position of the place where a user lives, we define a set of initial assumptions, as follows:

1. The position of the user is known as a consecutive set of readings from a position sensor (such as a GPS receiver or a positioning service in a cellular mobile network). However, it is not required that the user is being constantly tracked - there can be multiple gaps between the consecutive position readings, as happens when a user moves in a town with high buildings (urban canyons) or when entering inside buildings.

2. Every position reading is represented by a pair of coordinates (latitude, longitude) in the WGS84 datum, and a timestamp *ts* representing the instant of the reading.

In the following analysis, we denote by $P=\{p_1, p_2, \ldots, p_i, \ldots\}$ the unbound list of all position readings, where $p_i=((lat_i, long_i), ts_i)$ is the most recent reading and p_{i-1} is the previous reading.

In the subsequent analysis, although the position readings are treated as a list, the processing is performed sequentially as new readings are available and no storage of past readings is required except for the two most recent ones. In a real situation, only the two last readings need to be stored by the estimation process. Among other aspects, this approach contributes to the user privacy since, at any given instant, the position of the user in the past is unknown. Although this is irrelevant if the estimation process is to run on the user's mobile device, it could be important if the process is performed by a central server working on behalf of many users.

Probabilistic Model

The approach for estimating the "home place" is based on the assumption that the place where persons spend more time is at home and that their spatio-temporal behaviour is cyclic, that is, there is a periodic movement to and from the place where a person lives. We model this behaviour as a probability function that describes the *Probability of a user Be At Home*, P_{bah}, given the time of the day *dh*, in hours. This probability is model by the following simple sinusoidal function:

$$P_{bah}(dh) = \frac{1}{2}\left(1 + A_1 \times Cos\left(\frac{(dh - PeakHour)\times 2\pi}{24}\right)\right) \quad (1)$$

where A_1 and *PeakHour* are model parameters that represent the amplitude of the sinusoid and the hour of the day where the user is most probably at home, respectively. Since the value of *PeakHour* varies from person to person, it is considered a dynamic parameter and its value is estimated by the proposed algorithm.

This function is depicted in Figure 1.

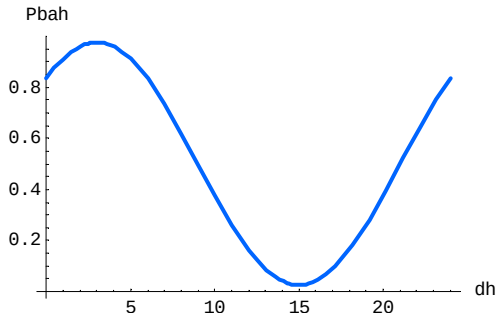


Figure 1: Probabilistic model for the presence of a person at home (A_1=0.95; *PeakHour*=3).

Given two consecutive position readings, we model the probability that a user stayed at home during the time that elapsed between the two readings as the average value of P_{bah} between this two time instants:

$$P_{bah}(dh_1, dh_2) = \frac{1}{dh_2 - dh_1} \times \int_{dh_1}^{dh_2} P_{bah}(dh) \cdot ddh = \\ = \frac{1}{2} - \frac{12 \times A_1}{2\pi \times (dh_2 - dh_1)} \times \left[Sin\left(\frac{\pi}{12} \times (dh_1 - PeakHour)\right) - Sin\left(\frac{\pi}{12} \times (dh_2 - PeakHour)\right)\right] \quad (2)$$

Although this is a very simple and empirical model, it has proved to describe the human behaviour well enough for the purpose of the analysis described below (see the Results section for the comparison of the model with real data). A similar model is used to describe the *Probability of a user Be At Work*, P_{baw}, given the time of the day *dh*. The mathematical definition of this model is similar to that in (1) with *PeakHour* replaced by *PeakHour-12*.

Position readings pre-processing

Since the original dataset can contain a number of collection gaps (time intervals without position data), some pre-processing is required to clean the dataset. From the initial list *P* of position readings, a list *Q* is created consisting of a sequence of data records that result from the pre-processing of the elements of *P* such that:

$$q_i = \begin{cases} \left(\left(\frac{(lat_{i-1} + lat_i)}{2}, \frac{(long_{i-1} + long_i)}{2} \right), ts_i, tp = ts_i - ts_{i-1}, d = Dist(p_{i-1}, p_i) \right), & \text{if not first point} \\ ((lat_i, long_i), ts_i, T_{STAY}, L_{RUN}), & \text{if first point} \end{cases} \quad (3)$$

where $Dist(p_{i-1},p_i)$ is the geographic distance between the two consecutive readings, and T_{STAY} and L_{RUN} are constants that are used to identify the beginning of a new track. The beginning of a new track is detected when the time that elapsed since the previous record is higher than T_{MIN} and, simultaneously, the distance to the previous position is higher than D_{MAX1}. In summary, Q represents the list of valid points, the amount of time the user spent at each position and the distance run between the current and the previous position.

In the second phase of the pre-processing, a subset R of the above data records is created, consisting on the points where the user stayed at least for some amount of time, i.e., the user has not moved significantly. These points represent the relevant data since we aim to estimate the position of places where the user stays for a long time (home and workplace). The data representing the travelling of the user between these places is not relevant. We include a point in this subset if the user stayed at that position for a period of time longer than T_{MIN} and the distance to the previous point is smaller than D_{MAX2}. List R is represented by the expression in (4).

$$R = \{r_1, r_2, ...r_i, ...\} = \{q_i = ((lat_i, long_i), ts_i, tp_i, d_i) \in Q : (tp_i > T_{MIN} \wedge d_i < D_{MAX2})\} \quad (4)$$

This new data subset is the input of the estimation algorithm described below.

Estimation algorithm

We start the estimation algorithm by creating a list of *Home Place Candidates* (*HPC*), $HPC=\{hpc_1, hpc_2, ..., hpc_N\}$, with N candidates for "home place", each of one with a data structure as follows:

$$hpc_j = (c_j, cf_j, lu_j, ast_j, hist_j) \quad (5)$$

where c=($cLat$, $cLong$) is the geographic *Centroid* of the candidate (the estimated position of the "home place"), cf (*Confidence*) is a measure of the degree of confidence calculated for this candidate, lu (*LastUpdated*) is the time instant when this candidate was last updated, ast (*AccumulatedStayTime*) is the total amount of time spent by the user in the location represented by the centroid, and $hist$ (*Histogram*) is an histogram of the amount of time spent at the location represented by the centroid per hour. A similar list is created for the *WorkPlace Candidates* (*WPC*).

Each of the N candidates is initialized with all its dimensions equal to zero. Once the "home place" and "workplace" candidates are initialized, each new reading, represented by a data record r_i, is processed by the **estimate**( ) function as described in Table 1.

Table 1: Estimation algorithm.

```
estimate()
1:  Assign a random value to the PeakHour parameter
2:  Read the first position fix
3:  while(true)
4:    Read a new position fix
5:    Pre-process the two most recent position readings to obtain r_i
6:    Store the current position fix
7:    Store the current list of "home place" candidates (oldHPC)
8:    Update HPC by clustering r_i: HPcluster(HPC, r_i)
9:    Update WPC by clustering r_i: WPcluster(WPC, r_i)
10:   if(ast of the first WPC element > ast of the first HPC element)
11:     Exchange the values of HPC and WPC
12:   if(HPC≠oldHPC)
13:     Estimate a new value for PeakHour (hour at which the histogram is maximum)
```

The **estimate**( ) function starts by assigning a random hour of the day to the *PeakHour* parameter. Alternatively, it could be assigned a value between 2 and 4 to *PeakHour*, since this is the time period at which most people are at home. Then, for each new position reading, the two most recent readings are pre-processed as described above, and the obtained register r_i is clustered into the list of "home place" candidates, *HPC*, and "workplace" candidates, *WPC*. Next, the values of the *ast* variable for the first *HPC* candidate and the first *WPC* candidate are compared. If the first *WPC* candidate accumulated a larger period of time than the first *HPC* candidate, it means that the first *WPC* candidate is locking into a position that has a great potential to be the home place. Therefore, the values of *WPC* and *HPC* must be exchanged. At the end, the value of the *PeakHour* variable is recalculated from the *HPC* list if this list has changed since the last iteration. The new value of this variable is calculated from the histogram of the most probable candidate (candidate number one) for the home place as the hour corresponding to the maximum value of the histogram.

The process of updating the list *HPC* of "home place" candidates (step 8 in Table 1) is based on a clustering algorithm and is described in Table 2.

Table 2: Clustering algorithm.

```
HPcluster(HPC, r_i)
1:  p_1 = the probability that the user has been at home given r_i
2:  if(p_1 > P_bahMIN)
3:    Calculate the list of distances to the centroids of all candidates
4:    Find the candidate nearest to the current position r_i
5:    Select the corresponding candidate (HPC_j)
6:    if(distance to the nearest candidate < D_LIM)
7:      Update the candidate j (recalculate c_j, and update lu_j, ast_j and hist_j)
8:    else
9:      Select the last candidate (HPC_N)
10:     if(tp_i > ast_N)
11:       Initialize candidate N (the cluster is formed by r_i only)
12: Update the value of the confidence value for the N candidates
13: Sort the HPC list in descending order of ast
```

This clustering process is implemented by the function **HPcluster**( ) where each new position record r_i is potentially assigned to one of the candidates. A position record r_i is assigned to one of the existing clusters ("home place" candidates) if two conditions are satisfied: (i) first, the probability that the user has been at home given the corresponding timestamp is higher than a pre-defined threshold P_{bahMIN}; (ii) second, it must be close enough to one of the clusters' centroids (at a distance lower than D_{LIM}). In this case, record r_i is clustered to the nearest candidate. This process is depicted in Figure 2.

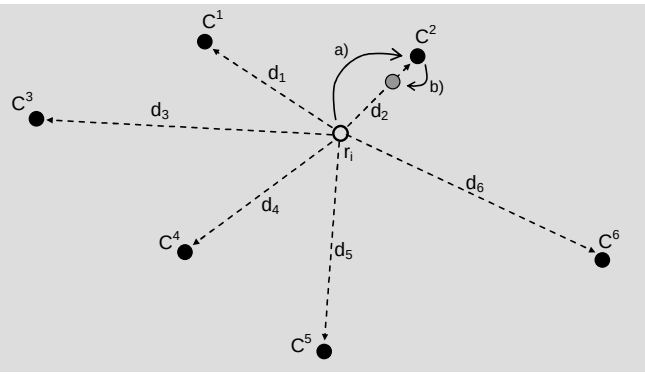


Figure 2: Clustering a new record: (a) The current record, r_i, is assigned to the closest candidate (candidate 2). (b) the centroid of this candidate is recalculated.

As a new record r_i is added to an existing cluster, the new position of the corresponding centroid, c_i, is calculated as the centre of gravity between the previous value of the centroid, c_{i-1}, and the current record r_i, where the masses are the total *AccumulatedStayTime* ast_{i-1} and the stay time of the current record tp_i, respectively (equation 6).

$$c_i = \left\{ \frac{cLat_{i-1} \times ast_{i-1} + lat_i \times tp_i}{ast_{i-1} + tp_i}, \frac{cLong_{i-1} \times ast_{i-1} + long_i \times tp_i}{ast_{i-1} + tp_i} \right\} \quad (6)$$

The position of the new centroid is equivalent to the centre of gravity of all the points that contributed to the candidate definition, and where the masses are the amount of time the user stayed at the corresponding positions. Therefore, the amount of displacement of the centroid from one iteration to the next tends to decrease unless the new point increases the spread of the contributing points. After a larger number of iterations, the displacement of the centroid will be very small, meaning that the true position of the "home place" was found (please see Figure 6).

If the second of the above conditions is not satisfied (distance lower than D_{LIM}), then the record r_i is used to initialize a new candidate (one of the initial *N* candidates), or to replace one of the already existing clusters if all *N* candidates have already been initialized. In the first case, the new cluster is formed by the record r_i itself. In the second case, one of the existing candidates is replaced, and initialized, if the user stayed at the current position an amount of time higher that the total *AccumulatedStayTime* of one of the existing candidates (the one with the lower value of *AccumulatedStayTime* is replaced). Otherwise, the record is discarded.

After each iteration (the processing of a data record), the candidate with the higher value of *AccumulatedStayTime* is considered to describe the most probable position of the "home place". Sorting the HPC list (step 13 in Table 2) places the best candidate at the top of the list.

In addition, a quality parameter, the *Confidence*, is maintained to indicate how good this estimation is. For each candidate *j*, this parameter is defined as the percentage of time a user spent at this candidate over the total amount of time spent in all active candidates, and is defined as:

$$cf_j = \frac{ast_j}{\sum_{k=1}^{N} ast_k} \quad (7)$$

where ast_j is the *AccumulatedStayTime* for candidate *j*.

The procedure for updating *WPC* is the same as used to update *HPC*. For the "workplace" candidates, the **WPcluster**( ) function is similar to **HPcluster**( ), except that P_{baw}() is used instead of P_{bah}().

RESULTS

In order to evaluate the proposed algorithms, we collected position data for a user during a period of about 3 months using a GPS receiver. Each position record includes the position of the user, measured as a pair (latitude, longitude) in the WGS84 datum, and a timestamp representing the instant in which the record was collected (date and time). We note that the collected data is not continuous and includes several periods of time without any data (collection gaps).

This data was partitioned into several time series and in this section we present the results obtained by processing some of these series.

For the following example we consider a data series collected over 26 consecutive days, with a total of 19865 collected position fixes.

After processing this data series with N=10 candidates, the algorithm estimated the position of each home place candidate and identified the most probable one. These results are shown in Figure 3 and 4, where: (i) the dotted line (blue) represents the user path; (ii) the position of each candidate is depicted as a solid circle (orange), with the numbers inside the circles representing where each candidate ranks in terms of *AccumulatedStayTime*. The values of the parameters of the algorithm used in this example were the following: A_1=0.95; T_{STAY}=5s; L_{RUN}=100km; N=10; D_{MAX1}=1000m; T_{MAX}=120s; T_{MIN}=600s; D_{MAX2}=500m; P_{bahMIN}=0.5; P_{bawMIN}=0.5; D_{LIM}=500m.

The candidate ranked number one is correctly located at the position where the user lives, thus confirming that the algorithm converged to the expected result. In this case, 7 of the 10 candidates were initialized by the algorithm as this data series includes many episodes where the user was considered to be at home. In Figure 4 an expanded part of Figure 3 is shown, where the candidate ranked number one is clearly shown (the candidates are ranked by the descending order of their *ast* values).

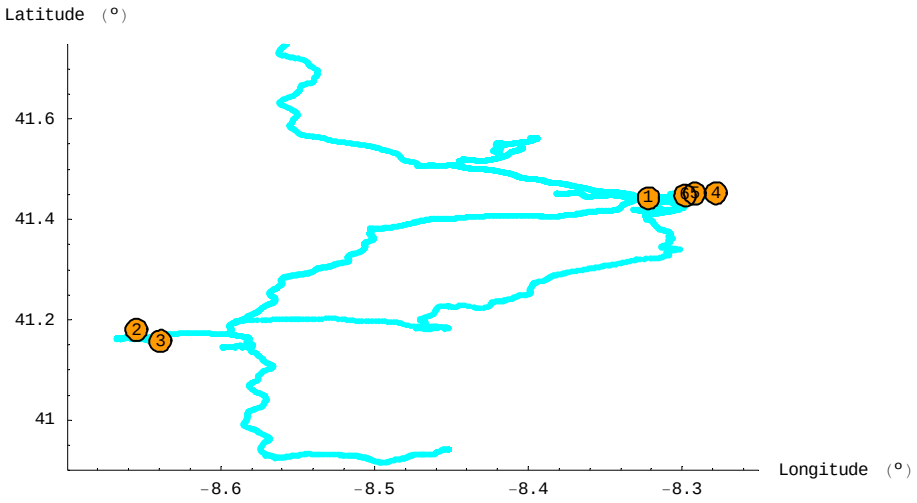


Figure 3: Estimated "home place" candidates.

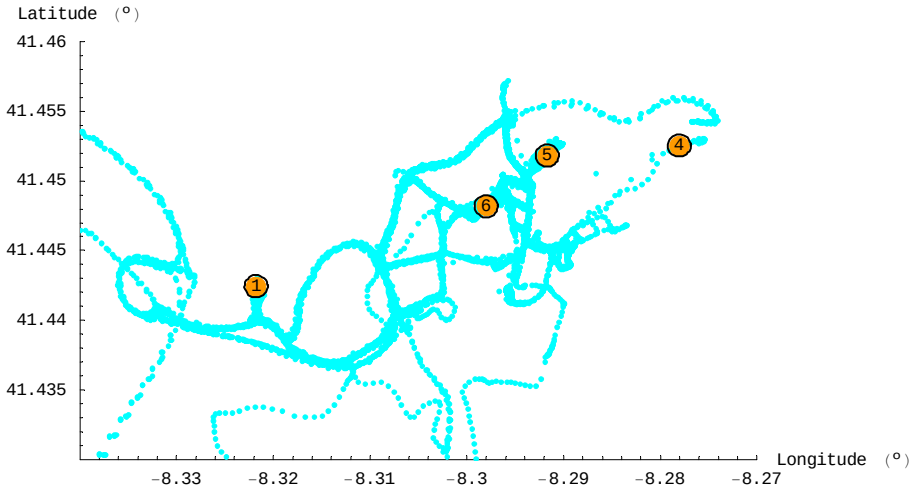


Figure 4: Estimated "home place" candidates (detail).

Figure 5 shows the results obtained on the estimation of the workplace for the same dataset. From the initial 10 workplace candidates, 8 were initialized (6 shown in Figure 5). The candidate ranked number one by the algorithm is actually the place where the user works most of the time (the University Campus). Candidate number 2 points to the user home, where the user also works frequently, and candidates 3 and 4 point to two other locations that are used occasionally as workplaces (other University facilities). In summary, the algorithm estimated a list of places that represent the actual places where the user works and ranked them in terms of relevance.

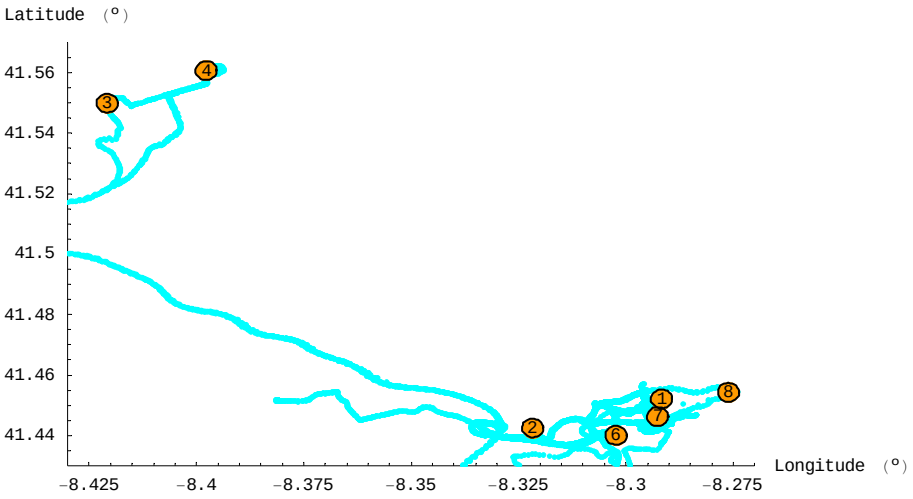


Figure 5: Estimated workplace candidates (partial view).

Temporal evolution of the Centroid parameter

As the processing of each new record by the proposed algorithm progress, the centroid's position for the several candidates (clusters) also evolves. For the candidate ranked number one at any particular moment, the position of the centroid changes when: (i) the first candidate in the list changes, and; (ii) every time a new record is assigned to a candidate (cluster). Figure 6 shows the evolution on the position of the centroid for the candidate that was ranked first at the end of the process (after processing the 19865 records).

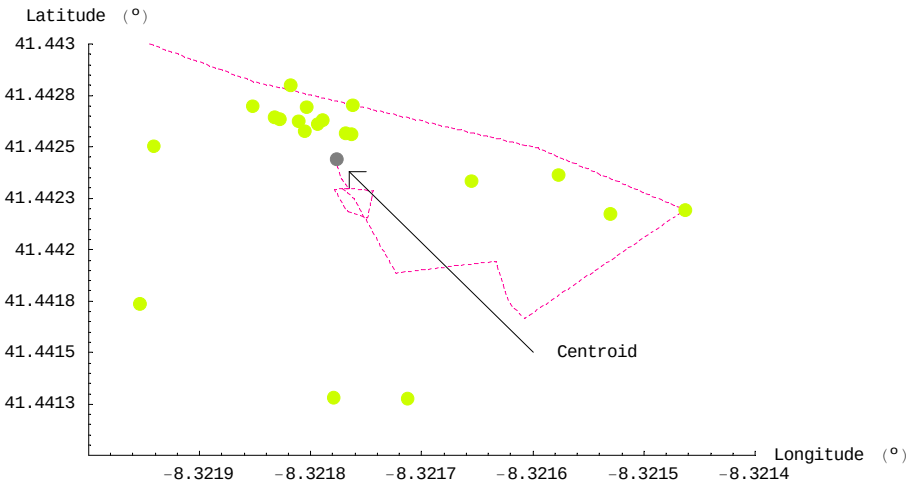


Figure 6: Temporal evolution of the centroid position.

In this figure, the solid green circles represent the position of the records that contributed to the calculation of the centroid, the gray circle represents the final position of the centroid and the red dashed line shows the temporal evolution of the centroid's position. As depicted in Figure 6, in each iteration the centroid moves into the direction of the new record (as depicted in Figure 2). The amount of displacement is proportional to the relative amount of time the user spent at that position. As more records are associated to a given cluster, the displacement of its centroid tends to decrease and the position of the centroid approaches a stable position. At the end, the position of the centroid represents the centre of mass of all contributing records.

Temporal evolution of the *PeakHour* parameter

In the previous example, the estimation process started with an initial random value of 11.3881 for the *PeakHour* parameter. As the estimation process evolved, the value of this parameter was dynamically adjusted. Figure 7 shows the temporal evolution of this parameter. In the initial phase of the clustering process, the value of the *PeakHour* parameter evolved from 11.3881 to higher values (13 and 23). Afterwards, as more position fixes were clustered to create the estimated "home place", the value of this parameter started to converge to its final value of 3 (3:00 am). This final value was attained after 12 iterations of the estimation process which, in this example, corresponds to 10 days of position data.

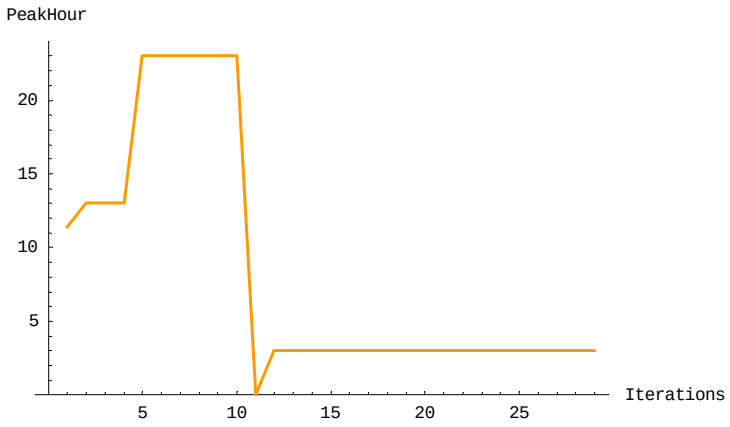


Figure 7: Temporal evolution of the *PeakHour* parameter.

Temporal evolution of the Confidence parameter

The *Confidence* parameter is a measure of how good is the estimated position of the home and workplace. As more records are processed, more information is available about the user spatio-temporal behaviour and more information is added to each of the candidates. One, therefore, expects the confidence on the estimates to increase as more records are processed. In Figure 8 we show how this parameter evolves as more records are added to a candidate. The values used in Figure 8 represent the confidence, as defined by equation (7), for the "home place" and "workplace" that were ranked in first and second position at the end of the process.

These results show that the algorithm rapidly converged to the correct estimate of the "home place". Regarding the estimation of the "workplace", the algorithm took a longer time to converge. In this case, the candidate that was ranked in second position at the end of the process has been ranked in first place for some time, being later replaced by another candidate. These "fluctuations" are due to the less regular spatio-temporal behaviour of the user in what concerns his presence at the workplace. Actually, the user considered in this example does not have a regular presence in a single place while working.

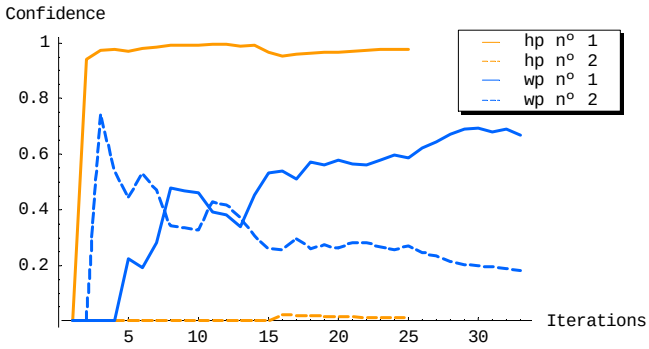
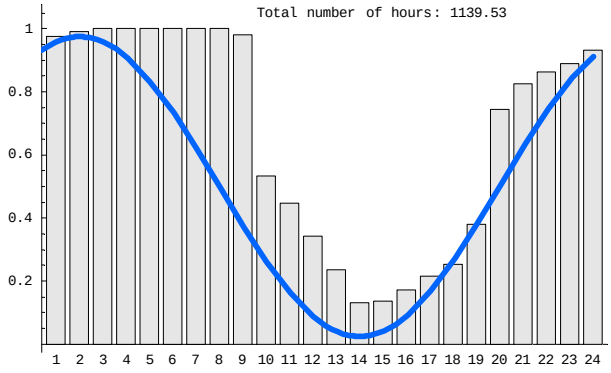


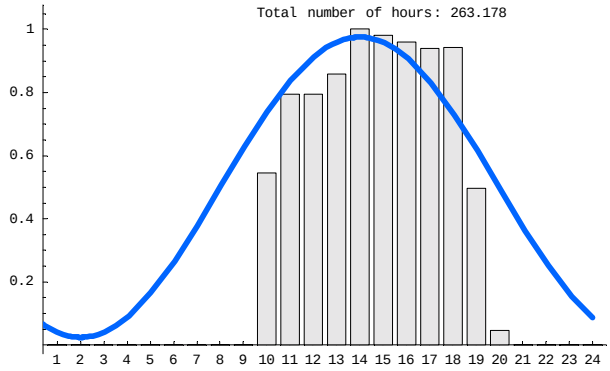
Figure 8. Evolution of the Confidence parameter while estimating the home and workplace.

Probabilistic model validation

To model the spatio-temporal behaviour of a person, we proposed a probabilistic model that describes the probability that a user is at home (or at work) given the time of the day. In this section we compare this model with actual data collected for one person. This comparison is shown in Figure 9, where the model curves are compared with a histogram of the permanency of the user at home (Figure 9a) and workplace (Figure 9b). These histograms were calculated from all the records that contributed to the definition of the most probable candidates (home and workplace respectively), from a data series comprising 91 days and more than 53 thousand position records.



a)



b)

Figure 9: Comparison between the probabilistic model and the real behaviour of a user: (a) while at home; (b) while at the workplace.

This comparison shows that, although simple, the proposed model describes the user behaviour quite well, in particular during the day. Note however, that the sum of the two histograms is not required to equal 1 for all hours since they only represent the temporal behaviour for two places and the user has visited many other places. These results also show that, in the analysed period, the user spent much more time at home (1139 hours) that at the more relevant workplace (263 hours). This data confirms our initial assumption that home is the place where most people spend more time. Further work on this model includes its validation for other persons with different temporal behaviours.

Sensibility of the clustering algorithm to the parameters values

All the previous results were achieved with a single set of clustering parameters. Most of these parameters are used during the pre-processing of the original data where the data series are cleaned from collection gaps and from temporal inconsistencies due to collection errors. Among the other parameters, P_{bahMIN} and P_{bawMIN} are the most relevant ones. The value of these parameters determines if a position fix is clustered into a candidate or not. In this section we analyse the sensibility of the clustering algorithm to variations on the values of these parameters. For this, we estimated the home and workplace positions using several values for the clustering parameters P_{bahMIN} and P_{bawMIN}. The obtained results are shown in Table 3. The data series used in these calculations was collected over 22 consecutive days, with a total of 8472 position fixes. Longer data series are expected to produce better results as the algorithm has more time to converge.

Table 3: Sensibility of the algorithm to variations on the values of some parameters.

Parameters	Correctly estimated home?	Correctly estimated workplace?	Confidence (home)	Confidence (work)
P_{bawMIN}=0.4, P_{bahMIN}=0.4	Yes	Yes	0.964	0.492
P_{bawMIN}=0.5	Yes	Yes	0.964	0.564
P_{bawMIN}=0.6	Yes	Yes	0.964	0.550
P_{bawMIN}=0.7	Yes	Yes	0.942	0.525
P_{bawMIN}=0.8	No	No	-	-
P_{bawMIN}=0.4, P_{bahMIN}=0.5	Yes	Yes	0.978	0.492
P_{bawMIN}=0.5	Yes	Yes	0.978	0.564
P_{bawMIN}=0.6	Yes	Yes	0.978	0.550
P_{bawMIN}=0.7	Yes	Yes	0.962	0.525
P_{bawMIN}=0.8	No	No	-	-
P_{bawMIN}=0.4, P_{bahMIN}=0.6	Yes	No	0.963	-
P_{bawMIN}=0.5	Yes	Yes	0.963	**0.576**
P_{bawMIN}=0.6	Yes	Yes	0.963	0.574
P_{bawMIN}=0.7	Yes	Yes	0.958	0.489
P_{bawMIN}=0.8	No	No	-	-
P_{bawMIN}=0.4, P_{bahMIN}=0.7	Yes	Yes	**0.986**	0.564
P_{bawMIN}=0.5	Yes	Yes	**0.986**	0.564
P_{bawMIN}=0.6	Yes	Yes	0.919	0.574
P_{bawMIN}=0.7	Yes	Yes	0.829	0.398
P_{bawMIN}=0.8	No	No	-	-
P_{bawMIN}=0.4, P_{bahMIN}=0.8	Yes	Yes	0.980	0.557
P_{bawMIN}=0.5	Yes	Yes	0.980	0.557
P_{bawMIN}=0.6	Yes	Yes	0.924	0.574
P_{bawMIN}=0.7	Yes	Yes	0.829	0.398
P_{bawMIN}=0.8	No	No	-	-

These results show that:

1. A large variation on the value of P_{bahMIN} (0.4 – 0.7) does not have a great impact on the correct estimation of the home and workplace. The same is valid for the value of P_{bawMIN}.

2. The value of the Confidence metric is always higher on the estimation of the "home place" than on the estimation of the "workplace". This result is coherent with the previous results,

namely those shown in Figures 5 and 8.

3. The values for these parameters that produce results with higher values for the Confidence metric are between 0.5 and 0.7, with a slightly higher value for the P_{bahMIN} parameter.

From these results it can be concluded that the clustering algorithm is quite insensitive to the values of these parameters. On the other hand, it is also difficult to find the optimal set of values for these parameters. Although this optimum value could be found for a single user, it may not be the optimum for all users. In this respect, the observed insensitivity of the algorithm can be seen as a positive characteristic of the algorithm as average values of the parameters should be good enough for a wide range of users.

RESULTS DISCUSSION

The above results show that the proposed clustering algorithm effectively estimates the home and workplace from a sequence of position records, and by processing only two records at a time. Moreover, the algorithm converges in a reasonable amount of time, even for users with irregular spatio-temporal behaviours, and after finding the correct positions it locks into them and is not affected by a large variety of episodes (e.g. the user being far away from home for a few days or several gaps into the position collection process). For the example results shown in Figure 3, only the data corresponding to 11 days was necessary for the algorithm to converge to the correct position of the home place and never, after that, the home place position was incorrectly estimated. Considering that these 11 days include a wide variety of episodes, the achieved results are very satisfactory.

Additionally, the algorithm does not require the storage of a large number of past position records (only one past record is required), thus contributing to a better control over the user privacy. Moreover, due to its low memory requirements, the algorithm can run on mobile devices with small amounts of memory.

These results also show that a simple probabilistic model is sufficient to describe the spatio-temporal behaviour of a user, and that this model provides enough information for the algorithm to correctly estimate the home and workplace positions. However, this model is not as good to describe the user while staying at his workplace (there is a significant difference between the model and the measured histogram) as it is for the description of the user while staying at home. Even so, it showed to be good enough for the correct estimation of the workplace position. On the other hand, this model was only validated for a single person – other persons might not be as good described by this model. Examples of persons for which the model might not be good enough include those that do not work on a fixed location such as taxi drivers. One possible improvement can eventually be achieved by using a more complex probabilistic model.

The results described in this paper can also be discussed in terms of their utility. Instead of trying to automatically estimate where a user lives and works, why not just ask him? We believe that, like the parameters estimated by the process described in this paper, there are plenty of other questions that a context-aware system could ask to a person and, therefore, turn it into a very intrusive system. With this work we aim to avoid this intrusive behaviour. Moreover, the work described in this paper is only part of a larger project that aims to automatically estimate many parameters about the context of mobile users. Simply asking a user a lot of questions is not a solution.

Model usage

As the clustering algorithm estimates the home and workplace positions, the available results can be used to infer context information for the user. At any given instant, given the user position, it is possible to estimate if the user is at home or at his workplace. As an example, consider that the user was found to be in p_i=(41.4422°,-8.3218°) at 17h54. Since this position is near the estimated home place (15 m away from the centroid) for which *cf*=0.9635, and also near the third candidate for the

estimated workplace (20 m away from the centroid) for which *cf*=0.1086, one can infer that:

$$context = \begin{cases} at\ HOME, with\ probability\ p = P_{bah}(17h54) \times cf_{home} = 0.1727 \times 0.9635 = 0.1664 \\ at\ WORK, with\ probability\ p = P_{baw}(17h54) \times cf_{work} = 0.8413 \times 0.1086 = 0.0914 \end{cases}$$

and, therefore, the user is most likely to be at home.

Related work

In the past a few other authors have reported different approaches to estimate locations where a mobile user has already been. In [1], Ashbrook et. al describe an approach to estimate "significant locations" also from GPS readings, where a clustering algorithm based on the k-means algorithm is used to learn locations visited by the users. In comparison to this work, the work described in this paper has the following differences: (i) instead of trying to estimate the position of locations already visited by the user, this approach estimates the position of the places where the user lives and works; (ii) the above places are automatically classified as "home place" and "workplace" without any user intervention; (iii) to each of these locations a radius and a value of confidence are assigned, which indicates the quality of the estimation; (iv) the estimation is performed sequentially as the position readings are acquired (no previous collection of data is required) and, therefore, this approach can be run in real time. In [9], Marmasse et. al also describe a module that learns "salient locations" from a set of GPS readings. However, in this approach, the labelling of the learned locations must be done manually by the user, as in [1].

CONCLUSIONS

In this paper we proposed a clustering algorithm that estimates the position where a person lives and works from a series of position readings. This algorithm works in a completely automatic manner, does not require any input from the user, and can be implemented to work in real time. The obtained results show that the algorithm is effective in estimating a set of ordered candidate positions, each one characterized by a centroid position and a confidence value. *A posteriori*, given the position of the user, one can infer if the user is at home (or workplace) with a certain probability if the distance between that position and each candidate is shorter than a specified threshold.

Further developments of this work include the validation of the approach for a broader range of users that include users with different spatio-temporal behaviours.

ACKNOWLEDGMENTS

This work was developed as part of the LOCAL project (http://get.dsi.uminho.pt/local) funded by the Fundação para a Ciência e Tecnologia through grant POSI/CHS/44971/2002, with support from the POSI program.

REFERENCES

1. Ashbrook, Daniel and Thad Starner, "Using GPS to learn significant locations and predict user movement across multiple users", Personal and Ubiquitous Computing, Volume 7, Number 5, October 2003.

2. Dey, Anind K., "Understanding and using context", Personal and Ubiquitous Computing Jornal, Volume 5 (1), Springer, 2001.

3. Ester, M., H.-P. Kriegel, J. Sander, and X. Xu, "Clustering for Mining in Large Spatial Databases", KI-Journal, Special Issue on Data Mining, 1, (1998), 18-24.

4. Ester, M., H.-P. Kriegel, J. Sander, and X. Xu, A Density-Based Algorithm for Discovering Clusters

in Large Spatial Databases with Noise, Proc. of the 2nd. International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, AAAI Press, 1996.

5. Fayyad, U.M., et al., eds. Advances in Knowledge Discovery and Data Mining, The MIT Press: Massachusetts, 1996.

6. Han, J., and M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers, 2001.

7. Kaufman, L. and P.J. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis. 1990: John Wiley & Sons, Inc.

8. Marmasse, Natalia and Chris Schmandt, "A user-centered location model", Personal and Ubiquitous Computing, Volume 6 (5/6), Springer, 2002.

9. Marmasse, Natalia and Chris Schmandt, "Location-aware information delivery with comMotion", HUC 2000 Proceedings, pp.157-171, Springer-Verlag.

10. Meneses, Filipe and Adriano Moreira, "A flexible location-context representation", The 15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, Barcelona, Spain, 5-8 September 2004.

11. Nurmi, Petteri, "Bayesian classifiers for context-aware computing", Research Themes in Context-Aware Computing – seminar, Department of Computer Science, University of Helsinki, Finland, January 2004.

12. Patterson, Donald J., Lin Liao, Dieter Fox and Henry Kautz, "Inferring high-level behaviour from low-level sensors" UbiComp 2003, The Fifth International Conference on Ubiquitous Computing, Seatle, Washington, USA, October 2003.

13. Schmidt, Albrecht, Kofi Asante Aidoo, Antii Takaluoma, Urpo Tuomela, Kristof Van Laerhoven, and Walter Van de Velde, "Advanced interaction in context", 1th International Symposium on Handheld and Ubiquitous Computing (HUC99), Karlsruhe, Germany, 1999.

14. Schmidt, Albrecht, Michael Beigl and Hans-W. Gellersen, "There is more to Context than Location", Computers & Graphics, Volume 23, Issue 6, December (1999).

15. Weiser, M.: The computer for the 21st century. Scientific American, Vol. 265, n.3 (1991) 94-104.

16. Yao, X., Research Issues in Spatio-Temporal Data Mining (White Paper), Workshop on Geospatial Visualization and Knowledge Discovery, University Consortium for Geographic Information Systems, Virginia, 2003.

AUTHORS INFORMATION

Adriano MOREIRA
adriano@dsi.uminho.pt
Universidade do Minho,
Portugal

Maribel Yasmina SANTOS
maribel@dsi.uminho.pt
Universidade do Minho,
Portugal